

Preaccelerator HV Auto-Reset

Local application

Thu, Feb 18, 1993

This local application provides an automatic reset of certain trip conditions that occur in the preaccelerator Haefely high voltage power supplies. If too many trips occur, it suspends its activity pending manual reset of the trip condition.

Up to six trip conditions can be specified as channel# parameters for this local application. The program monitors any of these trip conditions for being asserted. When any trip occurs, it waits 5 seconds before resetting the trip condition. If, during this 5 second delay, the trip condition vanishes (was manually reset), then the logic returns to await another trip condition. When the program issues the trip reset, it then turns on the HV power supply, and finally turns on the reference source. It then returns to await another trip condition.

If more than 3 trip conditions occur within one minute, the program enters a state of awaiting manual reset. A human must intervene. When trips again appear cleared, then it returns to await the next trip condition without requiring a disable/enable.

When the local application is first enabled, and there is already a trip condition present, it enters the same state of awaiting manual reset. Whenever it is in this state, it sets a Bit that is one higher than the enable Bit#. This Bit may be monitored for an alarm condition. It means the automatic reset logic is suspended awaiting manual reset of a HV trip condition. When the trip condition is cleared, this Bit is cleared also. If the local application is disabled, this Bit is also cleared.

Here are the parameters of the HVPS local application for the H- high voltage system:

```
E LOC APPL PARAMS 02/18/93 1354
NODE<0610>  NTRY< 8>
NAME=HVPS  CNTR=0000
TITL"HV PS AUTO-RESET      "
SVAR=00000000
ENABLE  B<00B0> I HVPS AUTO EN
LISTNUM  <000C>
STAT1   C<815B> IACOV
STAT2   C<815C> IANPS
STAT3   C<815D> IPSOC
STAT4   C<815E> IPSOV
STAT5   C<0000>
STAT6   C<0000>
HVPS ON C<0159> IHVPS
REFS ON C<0158> IREFS
```

Four status bits are checked for trip conditions. They are implemented as associated status bits of four analog channels. (The reason for using these pseudo channels to reference these trip status bits is that the channel# also has an associated control bit that

associated trip status. In this case, these channels are using the right-hand state as the active state. For example, the channel named HACOVS shows the state of the H- HV A.C. overload trip, and it exhibits the “OV” state in the right field. This is assumed for all the trip condition bits (represented by pseudo channel#s). An interrupt under the active OV state clears the trip. When the program performs this action, it does so by “interrupting” under that right-hand state. (For these 4 cases shown, however, there is only a single reset control action that will result from either the right-hand or the left-hand side.)

For the two pseudo channels that provide access to on/off control of the HV power supply and on/off control of the reference source, the active state is assumed to be the left-hand state. Their text reads “ ONOFF”, so the ON is represented by the left-hand state. This left-hand state is assumed by the program to be the ON state. (In these cases, a right-hand state control action would cause the supply to turn off.) At this writing, the sign bit of these two channel numbers has no significance.

Note: The active state of an associated status bit depends upon both the hardware active state and the invert bit for that bit in the corresponding analog descriptor for the channel. In the 4 cases given, the invert bit is zero, so that the active state of the associated status bit matches the hardware state.

Details

This local application uses a data request to collect the status information for monitoring purposes. The data request is a local one, of course. The list# parameter is needed to prevent collision with another list# in use, either by another instance of this local application (for H-, say) or any page application. Since page applications traditionally only use list#s in the range 0–7, list#s needed by local applications can use values in the range 8–13. This limitation can be removed, but it was not done by the time this application was written. This temporary “kludge” should be replaced by a method of dynamic assignment of list#s to avoid conflicts.